Advanced Data Management (CSCI 490/680)

Provenance

Dr. David Koop
Assignment 5

- Work with time series & spatial data
- Shorter assignment
- Cleaning, spatial rollup, rolling average
- Due April 30
- Questions?
Exam and Review

• Exam
  - Tuesday, May 5 from 4-5:50pm
  - Online
  - Similar format to Test 2
  - Comprehensive but with more focus on last few weeks of class
  - Information online soon

• Review
  - Thursday, April 30
  - Submit questions via email or discussion
Provenance in Computational Science

Data Management

Computation

Provenance

Visualization

Publishing
Provenance Questions

• What process led to the output image?
• What input datasets contributed to the output image?
• What workflows create an isosurface with isovalue 57?
• Who create this data product?
• When was this data file created?
• Why was \texttt{vtkCamera} used?
• Why do two output images differ?
Provenance & Causality

• Knowing what data/steps influenced other data/steps is important!
• Data dependencies: this output file depended on this input file
• Data-process dependencies: this output figure depended on these processes
• Causality can often be represented as a graph where connections represent dependencies
Provenance Capture Mechanisms

- **Workflow-based**: Since workflow execution is controlled, keep track of all the workflow modules, parameters, etc. as they are executed.

- **Process-based**: Each process is required to write out its own provenance information (not centralized like workflow-based).

- **OS-based**: The OS or filesystem is modified so that any activity it does is monitored and the provenance subsystem organizes it.

- **Tradeoffs**:  
  - Workflow- and process-based have better abstraction.  
  - OS-based requires minimal user effort once installed and can capture "hidden dependencies".
Abstraction: Script, Workflow, Abstract Workflow

data = vtk.vtkStructuredPointsReader()
data.SetFileName('head.120.vtk')

contour = vtk.vtkContourFilter()
contour.SetInput(data.GetOutput())
contour.SetValue(0, 67)

mapper = vtk.vtkPolyDataMapper()
mapper.SetInput(contour.GetOutput())
mapper.ScalarVisibilityOff()

actor = vtk.vtkActor()
actor.SetMapper(mapper)

cam = vtk.vtkCamera()
cam.SetViewUp(0, 0, -1)
cam.SetPosition(745, -453, 369)
cam.SetFocalPoint(135, 135, 150)
cam.ComputeViewPlaneNormal()

ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(cam)
ren.ResetCamera()
renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)

style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
iren.Start()
Abstraction: Script, Workflow, Abstract Workflow

```python
data = vtk.vtkStructuredPointsReader()
data.SetFileName('../examples/data/head.120.vtk')
contour = vtk.vtkContourFilter()
contour.SetInput(data.GetOutput())
contour.SetValue(0, 67)
mapper = vtk.vtkPolyDataMapper()
mapper.SetInput(contour.GetOutput())
mapper.ScalarVisibilityOff()
actor = vtk.vtkActor()
actor.SetMapper(mapper)
cam = vtk.vtkCamera()
cam.SetViewUp(0, 0, -1)
cam.SetPosition(745, -453, 369)
cam.SetFocalPoint(135, 135, 150)
cam.ComputeViewPlaneNormal()
ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(cam)
ren.ResetCamera()
renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)
style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
iren.Start()
```
Abstraction: Provenance Views
Prospective and Retrospective Provenance

- Recipe for baking a cake versus the actual process & outcome
- Prospective provenance is what was specified/intended
  - a workflow, script, list of steps
- Retrospective provenance is what actually happened
  - actual data, actual parameters, errors that occurred, timestamps, machine information

- Do not need prospective provenance to have retrospective provenance!
PROV: Three Views of Provenance

[Moreau et al., 2014]
Provenance-Enabled Systems

Table 1. Provenance-enabled systems.

<table>
<thead>
<tr>
<th>System</th>
<th>Capture mechanism</th>
<th>Prospective provenance</th>
<th>Retrospective provenance</th>
<th>Workflow evolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redux</td>
<td>Workflow-based</td>
<td>Relational</td>
<td>Relational</td>
<td>No</td>
</tr>
<tr>
<td>Swift</td>
<td>Workflow-based</td>
<td>SwiftScript</td>
<td>Relational</td>
<td>No</td>
</tr>
<tr>
<td>VisTrails</td>
<td>Workflow-based</td>
<td>XML and relational</td>
<td>Relational</td>
<td>Yes</td>
</tr>
<tr>
<td>Karma</td>
<td>Workflow- and process-based</td>
<td>Business Process Execution Language</td>
<td>XML</td>
<td>No</td>
</tr>
<tr>
<td>Kepler</td>
<td>Workflow-based</td>
<td>MoML</td>
<td>MoML variation</td>
<td>Under development</td>
</tr>
<tr>
<td>Taverna</td>
<td>Workflow-based</td>
<td>Scufl</td>
<td>RDF</td>
<td>Under development</td>
</tr>
<tr>
<td>Pegasus</td>
<td>Workflow-based</td>
<td>OWL</td>
<td>Relational</td>
<td>No</td>
</tr>
<tr>
<td>PASS</td>
<td>OS-based</td>
<td>N/A</td>
<td>Relational</td>
<td>No</td>
</tr>
<tr>
<td>ES3</td>
<td>OS-based</td>
<td>N/A</td>
<td>XML</td>
<td>No</td>
</tr>
<tr>
<td>PASOA/PreServ</td>
<td>Process-based</td>
<td>N/A</td>
<td>XML</td>
<td>No</td>
</tr>
</tbody>
</table>

[Freire et. al, 2008]
Provenance-Enabled Systems

Table 1. Provenance

<table>
<thead>
<tr>
<th>System</th>
<th>Storage</th>
<th>Query support</th>
<th>Available as open source?</th>
</tr>
</thead>
<tbody>
<tr>
<td>REDUX</td>
<td>Relational database management system (RDBMS)</td>
<td>SQL</td>
<td>No</td>
</tr>
<tr>
<td>Swift</td>
<td>RDBMS</td>
<td>SQL</td>
<td>Yes</td>
</tr>
<tr>
<td>VisTrails</td>
<td>RDBMS and files</td>
<td>Visual query by example, specialized language</td>
<td>Yes</td>
</tr>
<tr>
<td>Karma</td>
<td>RDBMS</td>
<td>Proprietary API</td>
<td>Yes</td>
</tr>
<tr>
<td>Kepler</td>
<td>Files; RDBMS planned</td>
<td>Under development</td>
<td>Yes</td>
</tr>
<tr>
<td>Taverna</td>
<td>RDBMS</td>
<td>SPARQL</td>
<td>Yes</td>
</tr>
<tr>
<td>Pegasus</td>
<td>RDBMS</td>
<td>SPARQL for metadata and workflow; SQL for execution log</td>
<td>Yes</td>
</tr>
<tr>
<td>PASS</td>
<td>Berkeley DB</td>
<td>nq (proprietary query tool)</td>
<td>No</td>
</tr>
<tr>
<td>ES3</td>
<td>XML database</td>
<td>XQuery</td>
<td>No</td>
</tr>
<tr>
<td>PASOA/PreServ</td>
<td>Filesystem, Berkeley DB</td>
<td>XQuery, Java query API</td>
<td>Yes</td>
</tr>
</tbody>
</table>

[Freire et. al, 2008]
Provenance-Enabled Systems

Table 1. Provenance

<table>
<thead>
<tr>
<th>System</th>
<th>Storage</th>
<th>Query support</th>
<th>Available as open source?</th>
</tr>
</thead>
<tbody>
<tr>
<td>REDUX</td>
<td>Relational database management system (RDBMS)</td>
<td>SQL</td>
<td>No</td>
</tr>
<tr>
<td>Swift</td>
<td>RDBMS</td>
<td>SwiftScript</td>
<td>Yes</td>
</tr>
<tr>
<td>VisTrails</td>
<td>RDBMS and files</td>
<td>YeML, a scripting language, allows users to use YeML to construct query</td>
<td>Yes</td>
</tr>
<tr>
<td>Karma</td>
<td>RDBMS</td>
<td>Karma</td>
<td>Yes</td>
</tr>
<tr>
<td>Kepler</td>
<td>Files; RDBMS planned</td>
<td>Kepler</td>
<td>Yes</td>
</tr>
<tr>
<td>Taverna</td>
<td>RDBMS</td>
<td>Taverna</td>
<td>Yes</td>
</tr>
<tr>
<td>Pegasus</td>
<td>RDBMS</td>
<td>Pegasus</td>
<td>Yes</td>
</tr>
<tr>
<td>PASS</td>
<td>Berkeley DB</td>
<td>XQuery</td>
<td>No</td>
</tr>
<tr>
<td>ES3</td>
<td>XML database</td>
<td>XQuery</td>
<td>No</td>
</tr>
<tr>
<td>PASOA/PreServ</td>
<td>Filesystem, Berkeley DB</td>
<td>XQuery, Java query API</td>
<td>Yes</td>
</tr>
</tbody>
</table>

[Freire et. al, 2008]
Today: Two types of provenance

• Database Provenance
• Evolution Provenance
Database Provenance

• Motivation: Data warehouses and curated databases
  - Lots of work
  - Provenance helps check correctness
  - Adds value to data by how it was obtained

• Three Types:
  - Why (Lineage): Associate each tuple \( t \) present in the output of a query with a set of tuples present in the input
  - How: Not just existence but routes from tuples to output (multiple contrib.'s)
  - Where: Location where data is copied from (may have choice of different tables)

[Cheney et al., 2007]
Provenance in Databases

A. Amarilli
Why Provenance

<table>
<thead>
<tr>
<th>Agencies</th>
<th>name</th>
<th>based_in</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁: BayTours</td>
<td>Santa Cruz</td>
<td></td>
<td>415-1200</td>
</tr>
<tr>
<td>t₂: HarborCruz</td>
<td>Santa Cruz</td>
<td></td>
<td>831-3000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ExternalTours</th>
<th>destination</th>
<th>type</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₃: BayTours</td>
<td>San Francisco</td>
<td>cable car</td>
<td>$50</td>
</tr>
<tr>
<td>t₄: BayTours</td>
<td>Santa Cruz</td>
<td>bus</td>
<td>$100</td>
</tr>
<tr>
<td>t₅: BayTours</td>
<td>Santa Cruz</td>
<td>boat</td>
<td>$250</td>
</tr>
<tr>
<td>t₆: BayTours</td>
<td>Monterey</td>
<td>boat</td>
<td>$400</td>
</tr>
<tr>
<td>t₇: HarborCruz</td>
<td>Monterey</td>
<td>boat</td>
<td>$200</td>
</tr>
<tr>
<td>t₈: HarborCruz</td>
<td>Carmel</td>
<td>train</td>
<td>$90</td>
</tr>
</tbody>
</table>

Q₁:
SELECT a.name, a.phone
FROM Agencies a, ExternalTours e
WHERE a.name = e.name AND e.type='boat'

Result of Q₁:
<table>
<thead>
<tr>
<th>name</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayTours</td>
<td>415-1200</td>
</tr>
<tr>
<td>HarborCruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

- Lineage of (HarborCruz, 831-3000):
  \{Agencies(t₂), ExternalTours(t₇)\}

- Lineage of (BayTours, 415-1200):
  \{Agencies(t₁), ExternalTours(t₅, t₆)\}

- This is not really precise because we don't need both t₅ and t₆—only one is ok

[Cheney et al., 2007]
How Provenance

- How provenance gives more detail about how the tuples provide witnesses to the result
- Prov of \((\text{San Francisco, 415-1200})\): \{\{t1\}, \{t1, t3\}\}
- \(t1\) contributes \textbf{twice}
- Uses provenance semirings (the "polynomial" shown on the right)

<table>
<thead>
<tr>
<th>Agencies</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>based_in</td>
<td>phone</td>
<td></td>
</tr>
<tr>
<td>t1:</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>415-1200</td>
</tr>
<tr>
<td>t2:</td>
<td>HarborCruz</td>
<td>Santa Cruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ExternalTours</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>destination</td>
<td>type</td>
<td>price</td>
</tr>
<tr>
<td>t3:</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>cable car</td>
</tr>
<tr>
<td>t4:</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>bus</td>
</tr>
<tr>
<td>t5:</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>boat</td>
</tr>
<tr>
<td>t6:</td>
<td>BayTours</td>
<td>Monterey</td>
<td>boat</td>
</tr>
<tr>
<td>t7:</td>
<td>HarborCruz</td>
<td>Monterey</td>
<td>boat</td>
</tr>
<tr>
<td>t8:</td>
<td>HarborCruz</td>
<td>Carmel</td>
<td>train</td>
</tr>
</tbody>
</table>

\(Q_2:\)

\[
\text{SELECT } a.\text{destination, } a.\text{phone} \\
\text{FROM } \text{Agencies } a, \\
(\text{SELECT name, } \text{based_in AS destination} \\
\text{FROM } \text{Agencies } a) \\
\text{UNION} \\
\text{SELECT name, destination} \\
\text{FROM } \text{ExternalTours } e \\
\text{WHERE } a.\text{name} = e.\text{name}
\]

\textbf{Result of } Q_2:\

\[
\begin{array}{ccc}
\text{destination} & \text{phone} & \text{polynomial} \\
\text{San Francisco} & 415-1200 & t_1 \cdot (t_1 + t_3) \\
\text{Santa Cruz} & 831-3000 & t_2^2 \\
\text{Santa Cruz} & 415-1200 & t_1 \cdot (t_4 + t_5) \\
\text{Monterey} & 415-1200 & t_1 \cdot t_6 \\
\text{Monterey} & 831-3000 & t_1 \cdot t_7 \\
\text{Carmel} & 831-3000 & t_1 \cdot t_8
\end{array}
\]

[Cheney et al., 2007]
Where Provenance

- Where provenance traces to specific locations, not the tuple values
- Q and Q' give the same result but the name comes from different places
- Prov of HarborCruz in second output: (t2, name)
- Important in annotation-propogation

### Agencies

<table>
<thead>
<tr>
<th></th>
<th>name</th>
<th>based_in</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>415-1200</td>
</tr>
<tr>
<td>t2</td>
<td>HarborCruz</td>
<td>Santa Cruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

### ExternalTours

<table>
<thead>
<tr>
<th></th>
<th>name</th>
<th>destination</th>
<th>type</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>cable car</td>
<td>$50</td>
</tr>
<tr>
<td>t4</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>bus</td>
<td>$100</td>
</tr>
<tr>
<td>t5</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>boat</td>
<td>$250</td>
</tr>
<tr>
<td>t6</td>
<td>BayTours</td>
<td>Monterey</td>
<td>boat</td>
<td>$400</td>
</tr>
<tr>
<td>t7</td>
<td>HarborCruz</td>
<td>Monterey</td>
<td>boat</td>
<td>$200</td>
</tr>
<tr>
<td>t8</td>
<td>HarborCruz</td>
<td>Carmel</td>
<td>train</td>
<td>$90</td>
</tr>
</tbody>
</table>

**Q₁:**

```
SELECT a.name, a.phone
FROM Agencies a, ExternalTours e
WHERE a.name = e.name
AND e.type = 'boat'
```

**Result of Q₁:**

<table>
<thead>
<tr>
<th>name</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayTours</td>
<td>415-1200</td>
</tr>
<tr>
<td>HarborCruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

**Q’₁:**

```
SELECT e.name, a.phone
FROM Agencies a, ExternalTours e
WHERE a.name = e.name
AND e.type = 'boat'
```

[Cheney et al., 2007]
Evolution Provenance
Data Exploration

[Modified from Van Wijk, Vis 2005]
Data Exploration

- Data analysis and visualization are iterative processes
- In exploratory tasks, change is the norm!

[Modified from Van Wijk, Vis 2005]
Exploration and Creativity Support

- Reasoning is key to the exploratory processes
- "Reflective reasoning requires the ability to store temporary results, to make inferences from stored knowledge, and to follow chains of reasoning backward and forward, sometimes backtracking when a promising line of thought proves to be unfruitful. ...the process is slow and laborious" — Donald A. Norman
- Need external aids—tools to facilitate this process
  - "Creativity support tools" —Ben Shneiderman
- Need aid from people—collaboration
Change-based Provenance: Photo Editing

• User Actions

• Undo/Redo History
Change-based Provenance: Photo Editing

- User Actions

- Undo/Redo History
Version Trees

- **Undo/redo stacks are linear!**
- **We lose history of exploration**
- **Old Solution:** User saves files/state
- **VisTrails Solution:**
  - *Automatically* & *transparently* capture entire history as a **tree**
  - Users can tag or annotate each version
  - Users can go back to **any** version by selecting it in the tree
VisTrails
VisTrails

- Comprehensive provenance infrastructure for computational tasks
- Focus on exploratory tasks such as simulation, visualization, and data analysis
- Transparently tracks provenance of the discovery process—from data acquisition to visualization
  - The trail followed as users generate and test hypotheses
  - Users can refer back to any point along this trail at any time
- Leverage provenance to streamline exploration
- Focus on usability—build tools for scientists
SAHM: Modeling the Spread of Invasive Species

[J. Morisette et al., USGS-Fort Collins, NASA]
UV-CDAT: Climate Science
UV-CDAT: Climate Science
UV-CDAT: Climate Science

[D. N. Williams, T. Maxwell, E. Santos, et al., LLNL, NASA, NYU]
ALPS: Large Quantum Simulations

[M. Troyer et al., ETH-Zurich]
Workflows

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPFile.url</td>
<td>web.mta.info/.../fares_130824.csv</td>
</tr>
<tr>
<td>CSVFile.skip_lines</td>
<td>2</td>
</tr>
<tr>
<td>JoinTables.left_col</td>
<td>STATION</td>
</tr>
<tr>
<td>JoinTables.right_col</td>
<td>_key</td>
</tr>
<tr>
<td>MplAxesProps.xlabel</td>
<td>Full Fares Purchased</td>
</tr>
</tbody>
</table>
Capturing Exploration: Version Tree of Workflows

```
HTTPFile
HTTPFile
CSVFile
JSONFile
JoinTables
ProjectTable
GMapCell

initial data
corrected data

station locations
station map
added fares
difference

november ff
november 2 data
August 16 Tab
full fares map

sum of ffs
30-D weekly
161st-River

broadway line
concourse line

get fare data (Group)
date range (Python source)
broadway diff map
```

D. Koop, CSCI 490/680, Spring 2020
Capturing Exploration: Version Tree of Workflows
Capturing Exploration: Version Tree of Workflows
Workflow Evolution Provenance
Workflow Evolution Provenance

- delete module “GMapCell”
- delete module “CellLocation”
- delete module “ProjectTable”
- delete module “SelectFromTable”

... 

- add module “SelectFromTable”
  - add parameter “float_expr” to “SelectFromTable” with value “latitude > 40.6”
- delete parameter “float_expr” from “SelectFromTable”
  - add parameter “float_expr” to “SelectFromTable” with value “latitude > 40.7”
- delete parameter “float_expr” from “SelectFromTable”
  - add parameter “float_expr” to “SelectFromTable” with value “latitude > 40.8”

...
Execution Provenance
Execution Provenance

<module id="12" name="vtkDataSetReader"
  start_time="2010-02-19 11:01:05"
  end_time="2010-02-19 11:01:07">
  <annotation key="hash"
    value="c54bea63cb7d912a43ce"/>
</module>

<module id="13" name="vtkContourFilter"
  start_time="2010-02-19 11:01:07"
  end_time="2010-02-19 11:01:08"/>

<module id="15" name="vtkDataSetMapper"
  start_time="2010-02-19 11:01:09"
  end_time="2010-02-19 11:01:12"/>

<module id="16" name="vtkActor"
  start_time="2010-02-19 11:01:12"
  end_time="2010-02-19 11:01:13"/>

<module id="17" name="vtkCamera"
  start_time="2010-02-19 11:01:13"
  end_time="2010-02-19 11:01:14"/>

<module id="18" name="vtkRenderer"
  start_time="2010-02-19 11:01:14"
  end_time="2010-02-19 11:01:14"/>

...
Parameter Exploration
Workflow Upgrades
Workflow Upgrades
Workflow Upgrades

[Diagram of a workflow process with nodes labeled CSVReader, StringToNumeric, AggregateData, JoinData, ExtractColumn, Scatterplot, Figure, FigureCell, AggregatedData, StringToNumeric, ComposeData, JoinData, ExtractColumn, Scatterplot, Figure, FigureCell.]
Workflow Upgrades

**Diagram 1:**
- CSVReader
- StringToNumeric
- AggregateData
- JoinData
- ExtractColumn
- MplScatterplot
- MplFigure
- MplFigureCell
- AggregateData

**Diagram 2:**
- CSVReader
- StringToNumeric
- AggregateData
- JoinData
- ExtractColumn
- MplScatterplot
- MplFigure
- MplFigureCell

**Diagram 3:**
- CSVReader
- StringToNumeric
- ComposeData
- JoinData
- ExtractColumn
- MplScatterplot
- MplFigure
- MplFigureCell
Provenance of Workflow Upgrades

Change-based Provenance:

delete connection StringToNumeric → AggregateData
delete connection AggregateData → AggregateData
delete connection AggregateData → JoinData
delete connection JoinData → ExtractColumn
delete connection JoinData → ExtractColumn
delete connection ExtractColumn → MplScatterplot
delete connection ExtractColumn → MplScatterplot
delete connection MplScatterplot → MplFigure
delete connection MplFigure → MplFigureCell
delete module AggregateData version 1.0.4
delete module AggregateData version 1.0.4
delete module ExtractColumn version 0.9.7
delete module ExtractColumn version 0.9.7
delete module MplScatterplot version 2.0.0
delete module MplScatterplot version 2.0.0
delete module MplFigure version 2.0.0
delete module MplFigure version 2.0.0
add module ComposeData version 1.1.0
add module ComposeData version 1.1.0
add module ExtractColumn version 1.0.2
add module ExtractColumn version 1.0.2
add module MplScatterplot version 2.0.1
add module MplFigure version 2.0.1
add module MplFigureCell version 2.0.1
add module MplFigureCell version 2.0.1
add connection StringToNumeric → ComposeData
add connection ComposeData → JoinData
add connection JoinData → ExtractColumn
add connection JoinData → ExtractColumn
add connection ExtractColumn → MplScatterplot
add connection ExtractColumn → MplScatterplot
...
Adding Provenance to 3rd-Party Tools
Adding Provenance to 3rd-Party Tools

Autodesk Maya

ParaView
Adding Provenance to 3rd-Party Tools
Adding Provenance to 3rd-Party Tools
VisTrails Provenance Plugin for ParaView
VisTrails Provenance Plugin for ParaView

[VisTrails, Inc.]
VisTrails Provenance Plugin for ParaView

[VisTrails, Inc.]
Querying and Re-using Provenance
Querying Provenance

- What process led to the output image?
- What input datasets contributed to the output image?
- What workflows include resampling and isosurfacing with isovalue 57?

- Graph traversal or graph patterns
  - How do we write such queries?
Querying Provenance by Example

- Provenance is represented as graphs: hard to specify queries using text!
- Querying workflows by example [Scheidegger et al., TVCG 2007; Beeri et al., VLDB 2006; Beeri et al. VLDB 2007]
  - WYSIWYQ -- What You See Is What You Query
  - Interface to create workflow is same as to query
Stronger Links Between Provenance and Data

- Filenames are often the mode of identification in data exploration
- We might also use URIs or access curated data stores
  - Always expected for exploratory tasks?
  - What happens if offline?
- Solution:
  - Managed store for data associated with computations
  - Improved data identification
  - Automatic versioning

[Koop et. al, 2010]
Provenance from Data

[Koop et. al, 2010]
Building Visualization Pipelines
Building Visualization Pipelines
Completions

[URL Completion, Safari]

[Code Completion, Intellisense]

[Web Search Completion, Google]
Visualization Pipeline Completions
VisComplete Overview

• Mine provenance collection: Identify graph fragments that co-occur in a collection of workflows (Data-Driven)
• Predict sets of likely workflow additions to a given partial workflow
Suggestion Interface
Suggestion Interface
VisComplete Results

[Diagram with arrows indicating user-added and completed sections]
VisComplete Results

- vtkDataSetReader
- vtkTransform
- vtkSphereSource
- vtkTransformFilter
- vtkRungeKutta4
- vtkStreamTracer
- vtkPolyDataMapper
- vtkActor
- vtkProperty
- vtkOutlineFilter
- vtkPolyDataMapper
- vtkActor
- vtkPolyDataMapper
- vtkActor
- vtkProperty
- VTKCell
- vtkMaskPoints
- vtkArrowSource
- vtkGlyph3D
- vtkMaskPoints
- vtkTubeFilter
- vtkPolyDataMapper
- vtkActor
-vtkRenderer

User-Added
Completed
Visualization by Analogy
Visualization by Analogy
Visualization by Analogy
Visualization by Analogy
Generating Visualizations by Analogy

A is to B as C is to D

PDB Report

<table>
<thead>
<tr>
<th>Protein Title</th>
<th>NEURAL CELL ADHESION MOLECULE, MODULE 2, HRR, 20 STRUCTURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors</td>
<td>P. H. JENSEN, V. BOROH, I. K. THOMSEN, V. BERÉZNIK, E. BOCK, P. M. POULSEN</td>
</tr>
<tr>
<td>Atom Count</td>
<td>9940</td>
</tr>
<tr>
<td>H</td>
<td>15440</td>
</tr>
<tr>
<td>N</td>
<td>2580</td>
</tr>
<tr>
<td>O</td>
<td>2880</td>
</tr>
<tr>
<td>S</td>
<td>60</td>
</tr>
</tbody>
</table>

PDB Entry
Generating Visualizations by Analogy

A is to B as C is to D
Generating Visualizations by Analogy

• Compute difference $\Delta(A,B)$ from provenance
  - $D = \Delta(A,B) \circ C$ is often not a valid workflow
Generating Visualizations by Analogy

- Compute difference $\Delta(A,B)$ from provenance
  - $D = \Delta(A,B) \circ C$ is often not a valid workflow
- Find map between A & C: $\text{map}(A,C)$
Generating Visualizations by Analogy

• Compute difference $\Delta(A,B)$ from provenance
  - $D = \Delta(A,B) \circ C$ is often not a valid workflow
• Find map between A & C: $\text{map}(A,C)$
• Compute mapped difference
  $\Delta AC(A,B) = \text{map}(A,C) \Delta(A,B)$
  - $D = \Delta AC(A,B) \circ C$
VisMashup

Acquire and Analyze Pipelines → Create Views (Simplify Pipelines) → Combine Views → App generation and deployment
VisTrails for Teaching Scientific Visualization

• “Using VisTrails and Provenance for Teaching Scientific Visualization” [Silva et al., Eurographics Educator Program, 2010]

• Same features that scientists use for exploratory tasks can also benefit students
  - Exploration: see all pipelines not just a “final” one
  - Comparison: see different pipelines and what changes exist
  - Assessment: see how a solution was developed
Provenance Analysis of Projects

Activity Histograms by Date

The data in the previous section shows that workflow evolution provenance analysis can help instructors to be more effective and improve the students' learning experience. Due to the provenance information, it is possible for one person to see what another person did, and to easily compare their own work to it. This makes it possible for the instructors to share their own work and to approach teaching by showing correct parameter values faster than others, and some users will also expend more effort tweaking parameters than others. Another interesting feature of these plots is that Task 5 shows a more direct path was used to obtain a solution. In contrast, there were branches to the second workflow. We found a range of branching factors that varied across users and tasks. For most tasks, this should be expected as some users love trial-and-error, and the branching factor of the version trees across the tasks for User 1 and User 2. A smaller branching factor indicates that the person did, and to easily compare their own work to it. This makes it possible for the instructors to share their own work and to approach teaching by showing how workflow evolution data can be used to understand the different types of work involved in a task.

Provenance evolution information can also be helpful to characterize tasks by the difficulty, due date, and how open-ended they were. To illustrate scientific visualization students varied in their goals, difficulty, due date, and how open-ended they were. To illustrate how workflow evolution data can be used to understand the different types of work involved in a task, we classified the actions involved in workflow development into:

- Parameter actions (modification of parameter values in the workflow);
- Structural actions (addition and deletion of modules and connections between modules in the workflow); and
- Layout actions (changes to the locations of modules in visual programming interface).

For all users, we calculated the overall percentage of actions that were structural, parameter types of actions involved. For all users, we calculated the overall percentage of actions that were structural, parameter types of actions involved. For all users, we calculated the overall percentage of actions that were structural, parameter types of actions involved. For all users, we calculated the overall percentage of actions that were structural, parameter types of actions involved. For all users, we calculated the overall percentage of actions.
Provenance Analysis of Projects

Comparing Paths to Solutions for Two Students

Branching Factor of Version Tree

User 1

User 2

Branching Structure for Task 3 of User 1

Branching Structure for Task 3 of User 2